Eur päisch s Patentamt	European Patent Office	Office européen des brevets

6B00/4095

EJU

REC'D 29 DEC 2000
WIPO   PCT

Bescheinigung	Certificate	Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.	The attached documents are exact copies of the European patent application described on the following page, as originally filed.	Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr.	Patent application No.	Demande de brevet n°

99308316. 1

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG,DEN
THE HAGUE,   08/11/00
LA HAYE,LE

EPA/EPO/OEB Form   1014   - 02.91

THIS PAGE BLANK (USPTO)

Europäisches
Patentamt

Eur pean
Patent Office

Office eur péen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: **99308316.1**
Demande n°:

Anmeldetag:
Date of filing: **21/10/99**
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):

BRITISH TELECOMMUNICATIONS public limited company

London EC1A 7AJ

UNITED KINGDOM

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

Resource allocation system

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

G06F9/46, G06F17/60

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

1

RESOURCE ALLOCATION SYSTEM

The present invention relates to methods and systems for allocating resources and can find application for instance in co-ordinating work, performed by either

5 machine resources or operatives, with the purpose of fulfilling overall work requirements.

Embodiments of the invention support for instance work co-ordination designed to take into account potential changes in resource availability. Such changes may arise for example as a result of changes in resource capacity, or working rates,

10 requests made by operatives and/or constraints set by local managers.

Where the resource concerned is directly machine-based, there can be several factors controlling use of the resources. All of these factors may change in real-time. There may be global constraints concerning a particular machine model, its maximum permitted working rate and changes in capacity enabled by new installations such as

15 supporting database capacity or other platform capabilities. There may be more local constraints, however still applying across several machines, such as noise control in specific geographical areas, or changes in available infrastructure. Clearly, there can be machine-specific factors such as machines overheating, going off-line or coming back on-line, or software problems blocking up part or all of a machine's capacity.

20 In a machine process environment, operatives may locally wish to use machine capacity for work in addition to a specified work requirement. The consequence of that local use of resource might however conflict with overall work limits for the best operation of the machine.

Most existing work control systems implement a process by co-ordinating a set

25 of activities that comprise a model of that process. Existing workflow systems use Petri Net-like models of work and rigidly enforce these models. Because of this dictatorial style they are often called "naziware" (Dourish et al. 1996).

Many workflow systems use a single "correct" process model controlled by a single person. Collaboration while planning a business process is supported by the

30 research system Regatta VPL (Swenson, K et al, 1994), and its commercial offsho t TeamWARE Flow. They allow business professionals to define and change "their" section of the process (their perspective) by manipulating a visual representation. However, it fails to support control and transparency across differ nt p rspectives.

2

According to a first aspect of the present invention, there is provided a method of supporting resource allocation in carrying out respective tasks which together fulfill one or more work requirements, each resource being provided with a resource interface, the method comprising:

5   storing constraint definition data defining constraints on said allocation of resources;

storing an initial data representation of resource availability;

receiving, from at least one of said resource interfaces, availability data concerning availability of a resource;

10   modifying a copy of said initial data representation according to said availability data to generate a proposed data representation of resource availability;

determining whether said proposed data representation is compatible with said constraint definition data;

in the case that said proposed data representation is compatible with said
15 constraint definition data, substituting said proposed data representation for said initial data representation to generate a new initial data representation; and

in the case that said proposed data representation is not compatible with said constraint definition data, generating and transmitting a rejection signal to said at least one of said resource interfaces.

20   Thus the method enables the allocation of task(s) to each resource such that the work requirements will be met and so as to accommodate availability data received via resource interfaces in the light of predetermined constraints.

The availability data can be entered to the system at the resource interface by operatives who may control equipment resources or who may themselves be
25 resources to which tasks are allocated. Alternatively, an equipment resource may itself enter availability data at the resource interface, generated for instance simply by failing or going offline. In this instance, an embodiment of the invention preferably further comprises, in the case that said proposed data representation is not compatible with said constraint definition data, generating and transmitting a failure
30 indication signal, for instance to a predetermined user interface or to resource allocation means.

An embodiment of the present invention can be used such that either machines or op ratives (ie peopl) can each make inputs (.g. via a r source interface including

3

or connected to a software agent) to an intermediary system which operat s to reconcile the inputs with (real or projected) constraints affecting an overall work project to be carried out,

Preferably, said constraint definition data comprises constraint definition data of
5   at least two types, a first type defining global constraints in respect of resources for carrying out the respective tasks which together fulfill one or more work requirements, and a second type defining constraints with respect to a specified set of resources, the method further comprising storing constraint definition data of said first type, receiving constraint definition data of said second type, and determining
10   whether said constraint definition data of said second type is compatible with the stored constraint definition data of said first type.

Embodiments of the present invention can thus be used to apply both global and local constraints, for instance both a local maximum task load for allocation t a specified resource and a global minimum task load for allocation to resources of a
15   specified type, and to review received local constraints against stored global constraints.

If the system is used for allocating operatives to carry out respective tasks towards an overall work requirement, then the global constraint definition data might define for instance statutory or company requirements in relation to workload while
20   the local constraint definition data might reflect policies for instance on overtim applied by individual team managers.

In an embodiment where the resources to be allocated are machine resources, then the global constraint definition data might relate for instance to a company policy that resources of a particular category should be allocated to tasks before
25   machine resources of a simpler or older category while the local constraint definition data might enable local conditions to be applied such as use of quieter machine resources at weekends.

This means that a local manager with authority over a plurality of resources or operatives may supplement the constraint definition data by constraints chosen by
30   him or her.

Local constraints which a local manager may wish to impose on a team of operatives as resources may b for instance constraints on the type of work to be

4

performed by ach operative. In a first example, a local manager may add local (environmental) data characterising the local geography for example (such as journey times between tasks). The data may be static (e.g. data defining the journey times between locations at which tasks are performed) or time-dependent (e.g. based on

5   present or projected weather or traffic conditions). In a second example, the local manager may be able to supply constraints relating to individual personnel (e.g. types of task for which individuals are qualified, and constraints relating to combinations of staff who have been found to work together well or badly in the past).

Where the resources are machines, equivalent constraints might arise from

10   using machines which are close together in a network, to keep network traffic down, or time-dependent in order to make best use of off-peak rates.

Preferably, the method further comprises storing, so as to be modifiable via a resource interface, data which is resource specific. This enables a further, important embodiment of the present invention. Preferably, in the case that said proposed data

15   representation is not compatible with said constraint definition data, the method comprises generating and transmitting a rejection signal to a plurality of said resource interfaces, each such resource interface being triggered to review its respective resource specific data and to output availability data concerning availability of its resource in dependence on the outcome of said review.

20   This means that each resource or operative can potentially be provided with, or set, a "personal profile" which is used by the resource interface allocated to that resource or operative to respond to rejection signals concerning other resources or operatives by outputting availability data. This feature makes the method interactive in respect of the resource interfaces without an operative or machine having to mak

25   real-time inputs in response to outputs or queries by the system. This interaction is important because the intermediary may transmit a rejection signal to a plurality of resource interfaces and receive an input from one of the plurality which changes th proposed data representation such that it becomes compatible with the constraint definition data. The intermediary can then retroactively accept an earlier input from a

30   resource interface which it would originally have rejected.

Typical inputs to the intermediary by the resourc interface, based on a personal profile, might relate to the type of work which the operative or resource do s, th order in which it is done, the time(s) at which it is done, and the identities

5

of one or more other of the operatives ("friends") with whom that operative wishes to cooperat . Thus, team-building activities are supported by nabling operatives to build a list of "friends" to help each other out. Team learning and knowledge sharing activities may be supported by using personal profiles to identify people who work on
5   the same task or in the same locality, and to provide information to "the expert" on a given issue.

This type of "personal profile", or resource profile, is also of use where the resource is a machine since it allows for instance a machine to identify a set of existing machines in a system for which it has compatible software installed and for
10 which it can substitute.

-A-further-possibility-where-resources-are-operatives or controlled by operatives, is for the method to determine (based on the tasks an operative is to perform, and their experience and qualifications) whether further explanation of the task will be required (or would be helpful). If so the explanation is transmitted to that operative.
15 Where the resource is a machine, this facility might be used to download data or software which the machine might need in order to fulfill a particular task.

An alternative which is also possible within the scope of the invention is for the method to accumulate instances of availability data, and at any time to determine a selection of instances which are compatible with constraint data based on the work
20 requirements, and to update the data representation according to that selection.

In the method described above, said one or more work requirements may be supplied by an operational support system. Global constraint definition data may also be supplied by the operational support system. Optionally, the operational support system may be adapted to supply work requirements for each of a plurality of local
25 managers, each of whom is responsible for a respective set (team) of resources. The operational support system may be thus able to apportion a large work project into a plurality of sections (one for each team), each having a number of work requirements. The operational support system may also supply local constraint definition data in respect of each respective set of work requirements, e.g. to the
30 local manager, for use in the method.

The method of the invention expressed abov can be carried out using a computer system referred to below in the specific description as an "intermediary". A local r team manager can b provided with an interfac for communicating with the

6

intermediary, and including data input means for inputting local constraint definition data.

It may be useful that a resource profile includes a priority indicator with respect to an availability commitment for its resource, or perhaps with respect to a type of
5   task. When the resource interface receives a rejection message, it can then review its resource profiles(s) for cases where a resource is committed to a task but with only low priority. For instance, the resource might be doing non-system tasks. Or the rejection message may be in respect of a task that the resource is defined to give top priority to. The resource interface may then output availability data in respect of
10 a resource which is already fully booked but which will jettison commitm nts as necessary if the availability data is accepted. This allows resources to provide selective back up in certain circumstances.

It may also be useful that a method according to an embodiment of the pr sent invention further comprises, subsequent to generating and transmitting a rejection
15 signal, triggering termination of tasks being carried out in respect of a common work requirement to which the rejection signal is related. This allows embodiments of th present invention to close down activity in support of a work requirement where it has emerged that changes in resource availability mean the overall work requir ment cannot be met. Preferably, there will be a time out before termination happens, to
20 allow resources to review their priorities.

According to a second aspect of the present invention, there is provided a system for supporting resource allocation in carrying out respective tasks which together fulfill one or more work requirements, the system comprising:

a data store for storing constraint definition data defining constraints on said
25 allocation of resources;

for each resource, a resource interface for inputting resource availability data;

an intermediary for receiving said resource availability data, reviewing it against stored constraint definition data and transmitting a message dependent on the r sult of the review to at least one resource interface;
30   at least one of said resource interfaces being provided with a resource specific data store, said r source interface being trigg rable by receipt of such a r sult-dependent message to review its resource specific data store and to transmit

7

r source availability data to the intermediary dependent on the outcom of the r view;

such that allocation of resources can be amended according to interaction between a resource interface and the intermediary, within limits determined by the
5 constraint definition data.

Allocation apparatus may determine an allocation of the work required to perform a work project between a plurality of such systems and transmit to each said system constraint data based on the required work to be performed by the respective resources available to that system. It is possible for the intermediaries of respective
10 systems to bid against each other (e.g. on behalf of their resources) to be allocated a certain section of the work. For this purpose the intermediaries may be provided with bidding functionality.

Embodiments of the present invention have particular strength in that they can support team-based collaboration in complex environments. For example call centre
15 staff could control call routing preferences, on-job training scheduling and so on.

An embodiment of the invention will now be described for the sake of example only with reference to the accompanying figures, in which:

Figure 1 shows schematically the functional structure of elements of a syst m according to an embodiment of the present invention;
20 Figure 2 shows schematically how the elements of Figure 1 may be supported by platform;

Figure 3 and 4 show schematically the primary components of a resource agent and an intermediary for use in the system shown in Figure 1;

Figure 5 shows a PC window displaying resource information in the
25 embodiment of Fig. 1; and

Figure 6 shows a PC window displaying intermediary information in the embodiment of Figure 1.

Referring to Figure 1, an embodiment of the present invention is illustrated schematically. The system includes an operational support system (OSS) 1, such as a
30 billing and work scheduling system. The OSS 1 is provided with, or generates, a definition of a work project to be carried out by a number of teams of operatives. For each team of operatives, there is an intermediary 3, in electronic communication with the OSS 1. Each operative is provided with an intellig nt r source interfac 5. Each

8

team has a manager, who is also supplied with an intelligent interface 7. Furthermore, there is an intelligent "trusted third party" interface 8 for use by a third party in arbitrating for instance between operatives and managers.

All of these functional blocks communicate using appropriate protocols. (The actual forms of communication are discussed below in more detail.)

Each resource interface 5 and the intermediary 3 can be considered to be a software agent in that each acts on behalf of an entity, has data or rules specific to the entity available to it, and can process incoming message content against the data or rules and output a response based on the outcome of the processing.

Referring to Figure 2, there are five primary types of data stored in the system. These are work project, or task, definitions 20, resource availability models 3b, global rules 3c, local rules 7b and resource profiles 5b.

An embodiment of the present invention is brought into use when the OSS 1 sends a definition of a work project to one or more intermediaries 3. The definition of a work project which is sent to each intermediary 3 will be in the form of a set of tasks for performance by the resources available to the respective intermediary 3, in this case by means of its team of operatives. The definition of the work project will be in the form of a schedule allocating resources to the tasks or sub-tasks. Th intermediary 3 transmits the scheduling information relevant to each operative to that operative's resource interface 5.

The scheduling of tasks received by the intermediary 3 from the OSS 1 is necessarily a "day 1" scheduling. It will be subject to amendment during performance of the tasks not least because of changes in resource availability from day to day. Where the resources are machines, this may be due to changes in installed versions, faults, updates, non-system workload and the like. Where the resources are controlled by operatives, the changes in availability may also occur as a result of the associated operative availability, for instance days off or preparedness to do overtime.

Having output the respective scheduling data to each resource interface 5, the intermediary agent 3 will receive data from any resource interface 5 for which there is a chang in availability, proposed or nec ssary. The intermediary 3 reviews the incoming availability data from the resource interfaces 5 against global rules 3c

9

and against local management rules 7b and accepts or rejects availability data from respective resource interfaces 5 in accordance with the outcom of the review.

The intermediary 3 for each team thus supports team co-ordination, and interfaces with the OSS 1. Relevant information is passed from the OSS 1 to the
5 intermediary. The intermediary 3 uses this information along with global and local rules 3c, 7b to support resources and/or operatives in interactive generation of a shared resource availability model. When the intermediary 3 has determined that an acceptable model has been generated, it passes the result back to the OSS 1 for use in its next evaluation cycle. During the evaluation cycle, the OSS 1 can use the
10 model for example to assess which workers are taking overtime so that pay can be varied accordingly, or to assess the quality of a given team or set of resources.

Thus the system provides means for implementing scheduling data from an OSS 1 in respect of resources to carry out a work project, while accommodating changes in availability of the resources without conflict arising over local
15 management rules 7b or global rules 3c. Hence day to day changes in resource availability are acceptable within a predetermined framework.

Each team manager can use their team manager interface 7 to input and modify local rules 7b for their team. A further function of the intermediary 3 is to review the local rules 7b against the global rules 3c which have priority. Hence th
20 intermediary 3 also provides a mechanism for team managers to set local rules 7b without breaking global rules 3c which may, for instance, derive from statutory requirements or corporate policy.

From the operatives' point of view, a particular strength of embodiments of the present invention lies in the provision of resource profiles 5b. Using their
25 resource profiles 5b, operatives can set workload preferences such as task difficulty levels, and build informal alliances to help each other with their requests. Parameters of this type can be implemented in use of the system by the resource interfaces 5. When a resource interface 5 receives scheduling information or notice of a change in availability data regarding another resource, the resource interface 5 reviews the
30 resource profile 5b in respect of its resource (or operative) and in certain circumstances may output availability data t its interm diary 3. This might occur in the following circumstances. A resource or operative may require to change availability, for instance by taking a day off or by being taken out of the service for

10

maintenance. This resource will effectively send a request via its resource interface 5 to its intermediary 3 for a change in availability status. If the intermediary 3, on review of the local rules 7b and/or the global rules 3c, finds that the change conflicts with a rule, the intermediary 3 will return a rejection to the relevant resource

5   interface 5, copied to each other resource interface 5 designated as being in the same team. One of the resource interfaces 5 receiving the rejection notice, on review of its resource profile 5b, may find the resource or operative being r jected listed there as subject to special status. This resource interface 5 may then issue effectively an offer to cover in place of the rejected resource. The intermediary 3 will

10   review the offer and issue an acceptance notice. The resource interface 5 which received the rejection notice may now resend its original request. The intermediary 3 will review the request in light of the offer and, as long as there is no conflict with the local 7b or global rules 3c, will now accept the request.

An operative can thus control their work environment by interacting via their

15   resource interface. Control can either be exercised at run-time (for example by selecting the next task) or at the planning stage (for example by specifying task difficulty preferences for the next day). Once an operative specifies his or her preferences, the resource interface will negotiate these preferences with the rest of the team through the intermediary 3.

20   (One resource interface 5 may serve more than one resource. It will n d access then to more than one resource profile.)

The intermediary 3 employs the local and global rules 7b, 3c to balance requests against expected workload for a team. The team manager can develop and change the set of local rules 7b available to the intermediary. This can be done by a

25   mechanism similar to setting the resource profiles. The manager has access via th manager's interface 7 to modify the local rules 7b, and the changes will be submitt d to the intermediary 3. The intermediary will check if they are compatible with the global rules 3c and output a rejection message if there is incompatibility.

The same mechanism can be expanded to cover multiple levels of authority, and

30   to allow different people to modify different aspects of the system.

Th system can provide support for team collaboration by notifying all resource interfaces 5 of the results of all requests, and the reasons for any rejection. This comparatively simple notification strategy enables a variety of negotiation and

collaboration activities when combined with the intelligent behaviour of the resource interfaces 5.

Figure 1 shows the functional communications paths set up between parts of the system in use. Referring to Figure 2, in order to support functionality as described
5 above, the system is structured as follows:

Processing software 3a, 5a, 7a, for the intermediary 3, the resource interface 5 and the team manager interface 7 is provided on a server 205 local to the team. The intermediary 3 and the interfaces 5, 7 have access to a database 210 for storing rules and models for both the intermediary 3 and the interfaces 5, 7. The database
10 210 may be supported by capacity on the server 205 itself, or elsewhere. Users —(operatives and —managers)—can—run—the -processing—software—3a, 5a, 7a- within appropriate limits by means of graphical use interfaces (GUIs) on their personal computers 5c, 7c. Processing software 8a for an independent manager agent or trusted third party 8 is also provided on the server 205, and a GUI on their personal
15 computer 8c. A computer supporting the OSS 1 is connected to provide work requirements 200 to the intermediary 3.

Communications among the platform elements mentioned above, the server 205, personal computers, database 210 and OSS platform, are provided by any suitable links, such as a local area network 215, adapted to carry the relevant
20 protocols, further described below.

The distribution of processes and data amongst platform in this arrangement may of course be modified, depending where capacity is or becomes available. For instance, the resource interface processes 5c may in practice be loaded on respective PCs 5c.

25 At a simple level the way the system can work day to day is that an OSS 1 establishes a resource model with an intermediary by sending data concerning resources in their team. For instance, if the resources are people, this will be a list of the people plus their current working status, such as "day off" or "working". Th intermediary builds a model of the team from the data.

30 The OSS then issues work requirements 200 to the intermediary 3. These work requirements may for instance be issued as a list of tasks or the intermediary may be equipped to translate a work requirement to a list of tasks. The intermediary 3 issues on or more r quests to its resource int rfac s 5 for offers of worktime

12

against tasks. Each resource interface 5 reviews the requests against the data it has stored for its respective resource (or operative), outputs an offer back to the intermediary 3 if the result of the review makes that appropriate, and updates its data to reflect the offer. The intermediary 3 builds a proposed model of resource

5   allocation against tasks, reviews the proposed model against the global rules 3c it has stored, and against local rules 7b set by team managers, installs the proposed model if it satisfies the rules, and issues notices to the relevant resource interfaces 5 that their offers have been accepted or rejected as appropriate.

The effect of this is that the intermediary 3 can resolve conflict and apply

10 both team (local) and global rules. Conflict can arise for instance because a team manager sets a new team rule, such as increasing the amount of overtime to be done per operative, and that new team rule conflicts with an existing global (eg company or legislative) rule. The intermediary 3 reviews any new team rule immediately against the global rules and notifies the relevant manager agent 7 of any conflict.

15 Operative preferences can also conflict with the rules. This will be resolved automatically by the intermediary 3 in use since it will simply reject offers and requests issued by a resource interface 5 which conflict with the rules, with an explanation.

Referring to Figures 1 and 3, taking the components of the system in turn, a

20 resource interface 5 comprises a GUI 5c, processing software 5a, data in the form of a resource profile 5b, and a message handler 5e of known type for sending and receiving messages using the known Agent Communication Language (ACL), compliant with the de-facto FIPA standard. In use, the resource interface 5 will also maintain a real time record 5d of current commitments, offers and rejections, in

25 respect of its resource.

The role of the resource interface 5 is to maintain the resource profile of a specified resource or operative, to maintain the real time record 5d, and to use the profile and real time record in negotiating with the intermediary 3 on behalf of the resource or operative. The resource interface 5 provides a GUI 5c for instanc by

30 downloading forms, drop down menus and the like, to the operative's personal comput r in order for th m to input the profile. This might include for instance the following:

o   Screen appearance

13

- Information filters
- User or resource groups ("friends")
- Task difficulty levels
- Working practices (such as "only pay back overtime owed to another operative"

5   or "only allocate tasks to this resource if a resource of another type is unavailable")

The resource interface 5 stores this profile 5b as a set of rules. The real time record 5d meanwhile records the resource's or operative's actual and proposed commitments. These are determined by interaction with the intermediary 3.

10   Referring to Figure 4, an intermediary 3 comprises OSS interface software 40, processing software 3a and, again, a message handler 3e for sending and receiving messages using the Agent Communication Language (ACL), compliant with the de facto FIPA standard. The message handler will be used by the intermediary 3 in communicating with the resource, team manager and trusted third party interfaces 5, 15 7, 8.

The role of the intermediary 3 is to maintain a model 3b of the current availability of its resources, to maintain the local and global rules 7b, 3c, receive availability messages from resource interfaces 5, review them against the local and global rules 7b, 3c and issue rejection messages, or acceptance messages, 20 appropriately to resource interfaces 5. It also receives task definitions 20 from the OSS 1, uses them to generate at least an initial model 3b of task allocation amongst resources, reviews an existing task allocation model 3b in the light of incoming availability messages from resource and interfaces 5b and outputs data on actual task performance to the OSS 1 to support management processes therein.

25   The following provides an example of the system in use, with examples of message content being passed between elements of the system. It is primarily given in terms of the resource interfaces 5 acting for operatives of resources to carry out tasks but it will be clear that the same principles will apply where the resource interfaces 5 are acting for resources themselves.

30

1.   **Setting up**

In order to set the system up, the OSS 1 provides data to the intelligent components in the system. For instance, the resource interfaces 5 will n ed data in

14

resp ct of their resources before they are equipped to act on behalf of those resources. The OSS 1 provides information such as:

1.1   Workers name details and working status

5   Eg   EmployeeInfo

{name: Fred Smith	person's name
Computer: 143.199.123.456	computer address
Job description: manager	engineer | manager
Workstatus: working}	working | day off | overtime

10

If not already instantiated, this data is used to instantiate the resource interfaces 5 in a team, and the team manager's interface 7, via the relevant intermediary 3. Depending on the circumstances, it may also be necessary that the OSS 1 provides time or date information so that availability information can be

15   mapped against time or date. The intermediary 3 itself then selects data with respect to availability of its resources and forms an initial "confirmed availability model":

1.2   Confirmed model:

{ListOfEmployees{all EmployeeInfo}

20   WorkersDayOff = 0   sum of employees taking day off

WorkersOvertime = 1}   sum of employees taking overtime

In order for the system to be enabled to respond in use on behalf of specific resources or operatives, the resource interfaces 5 need to be loaded with respective

25   resource profiles 5b. This can be done by operatives via the GUIs 5c. Alternatively, profiles could be loaded from an output of the OSS 1. This might be more appropriate for instance where the resource interface 5 is to act for a resource and the data involved in a resource profile 5b might comprise aspects such as communications protocols the resource is equipped to use, available processing

30   capacity and the like.

2.   **The method f r getting a day off**

15

An operativ (Ken) clicks the day off button on his GUI 5c. The resource interface 5 uses its processing software 5a and message handler 5e to translate this into the following agent message and sends it to the intermediary 3:

5   2.1   Request

(Request

    :sender KenAgent

    :receiver Intermediary

    :content :effect   Ken is an Employee who is seeking day off

10   :id 1

    :context 1

)

(Although not shown here, the request may of course also need to contain
15 time or date information if the request is relevant to a specific time or date.) The intermediary takes the confirmed model (1.2 above), adds 1 to the WorkersDayOff field and uses it as the proposed model:

2.2   Proposed model:

20   {ListOfEmployees{all EmployeeInfo}

       WorkersDayOff = 1

       WorkersOvertime = 1}

The intermediary 3 now checks the proposed model against the relevant local
25 and global rules 7b, 3c. The rule for taking a day off is that 2 employees need to b doing overtime before a person can take a day off.

2.3   Rule:

       if (WorkersDayOff*2 < WorkersOvertime)

30   DayOff = true

    Else

       DayOff = false

16

The proposed model fails the rule so the request is refus d and the proposed model abandoned. The refuse message below is sent to all the resource interfaces 5.

2.4   Refuse Message:
5   (Refuse

:sender Intermediary

:receiver PatAgent

:content   :ActionContent :effect   Ken is an Employee who is seeking day off
            :ReasonContent:

10   Fact is we are 2 short of `overtime' volunteers.

:id 2

:context 1

)

15   Ken's resource interface 5 will make the message available via the GUI 5c. Ken's resource profile may have a rule that in the event of a refuse message on a question of a day off, the resource interface 5 should resubmit the request after a fixed time period. Alternatively, the GUI 5c can be used to enter that information on receipt of the refuse message.

20   The other resource interfaces 5, on receipt of the refuse message will read the "ReasonContent" part of the refusal and, on reviewing the rules in their resource profiles 5b, may then issue a message to the intermediary 3 for at least two reasons. The resource profile 5b may for instance include a rule to request overtime in the event of receiving any message indicating that overtime is available. Alternatively, 25 the resource profile 5b may include a rule to request overtime in the event of receiving a refuse message against a day off request which identifies Ken or Ken's resource interface. This latter rule type provides strong flexibility in the system with respect to individual operators.

30 2.5   Simple Rule

If (R asonContent = ??? short of `overtime' volunteers)

Then (evaluateOvertimeRequest)

17

Evaluation of the overtime request may mean for example reviewing the real time record 5d of current commitments for a resource against the resource profile 5b in order to see whether the resource is already committed to overtime and whether the resource has a rule to do or not to do overtime in respect of "Ken". Depending

5 on the outcome of the evaluation, the other resource interfaces 5 can either ignore the message or request overtime. This can either be done automatically or on confirmation via a GUI 5c by the relevant employee. In the case that the outcome of the evaluation indicates that overtime should be offered, a request is sent to the intermediary 3.

10

2.5 Overtime Request

(Request

   :sender MelAgent

   :receiver Intermediary

15   :content :effect   Mel is an Employee seeking overtime

   :id 3

   :context 1

)

20   The intermediary 3 again generates a proposed model which will now be as follows:

2.6   Proposed model:

   {ListOfEmployees{all EmployeeInfo}

25   WorkersDayOff = 0

    WorkersOvertime = 2}

     The new proposed model will now be found to satisfy the rule for overtime. The intermediary 3 agrees to the overtime and the following message is sent to the

30 other employees.

2.7   Accept Message

(Agree

18

```
:sender Int rmediary
:receiver PatAgent
:content :ActionContent :Effect        Mel is an Employee seeking overtime
        :PreconditionContent
5               Done
:id 4
:context 1
)
```

10   The intermediary 3 now loads the proposed model as the current confirmed model which becomes:

```
2.8    New Confirmed Model
       {ListOfEmployees{all EmployeeInfo}
15         WorkersDayOff = 0
           WorkersOvertime = 2}
```

The original resource interface 5 (for Ken) can evaluate from the accept message that there is an increased chance of a day off. Ken's resource interface 5 20 (either automatically or by getting validation from a user) can then resubmit the original request for overtime. This time the request will be accepted.

In a system acting for resources instead of operatives of resources, an equivalent scenario for the one given above might arise for example where a s lf-diagnostic resource has detected an increased rate of faults occurring and issu s 25 notice of a requirement for (effectively a "request for") downtime so that it can reboot. If the requirement for downtime is then refused, only resource interfaces for resources which can offer equivalent functionality might then be triggered to offer increased processing capacity, enabling the faulty resource to go offline.

30 **3.   Local Rules**

A second function of the int rm diary 3 is to rec ive and validate local rules against global rules. This allows a team manager to add or modify a rule.

19

The manager's interface 7 provides a GUI which allows pre-existing local rules 7b to be manipulat d by pull down menus. For example, ther may a pre-existing rule that "x" people need do overtime for "y" people to take a day off, where x and y can be changed by the manager. (Where the resource interfaces 5 are acting 5 directly for resources, an equivalent rule might arise for instance where there's a need for a fixed amount of spare processing capacity to be available at all times. A request by a resource to use capacity for running a standby process instead of an allocated task might then be controlled by a local rule determining the relevant level of spare processing capacity in the team.) In order to change the rule, the manager 10 changes x or y and submits the new rule. The GUI submits the new rule in the form of a request as follows.

3.1   New Rule Request
(Request
15   :sender PatAgent
        :receiver Intermediary
        :content :effect   OvertimeRule dayOff y Overtime x
        :id 1
        :context 2
20)

The intermediary 3 reviews the request against the stored global rules 3c. If the outcome of the review is positive, the local rule 7b for that manager's team is changed.

25

3.2   Modified Rule
if (WorkersDayOff*x < WorkersOvertime*y)
        DayOff = true
    Else
30   DayOff = false

The intermediary will also send an agreement message to the GUI for the relevant manager.

20

3.3　　　Agr ement Message

(Agree

　　:sender Intermediary

5　　:receiver PatAgent

　　:content :ActionContent :Effect OvertimeRule dayOff y Overtime x

　　　　　:PreconditionContent

　　　　　　Done

　　:id 2

10　　:context 2

)

If a completely new rule is required, it is possible for the GUI to offer the manager free text input, such as "only give Fred days off". Alternatively, the GUI

15 can offer a visual rule builder. A new rule request is then submitted by the GUI to the intermediary 3 as before.

3.4　　New Rule Request

(Request

20　　:sender PatAgent

　　:receiver Intermediary

　　:content :effect　　NewRule only give Fred days off

　　:id 1

　　:context 2

25　)

In this case, because the new rule has been entered in free text and is not a modified version of an existing rule type, available via a pull down menu, the intermediary 3 sends the new rule request to an interface 8 for a more senior

30 manager for verification.

3.5　　Rule Verification Request

(Request

21

:sender Intermediary

:receiver SeniorManager

:content :effect   VerifyRule only give Fred days off

:id 1

5   :context 2

)

    The senior manager can then either refuse this rule or agree with it. If refusing, a refuse message is sent which the team manager's interface 7 makes

10   available to the manager.

3.6   Rule Refusal Message

(Refuse

    :sender SeniorManager

15   :receiver Intermediary

    :content :ActionContent :effect VerifyRule only give Fred days off

        :ReasonContent

        Rule not legal.

)

20

    If on the other hand the rule is agreed then the Senior Manager via a user interface 8 will enter the rule as a new local rule 7b, in a machine understandable form, and send an agreement message to the manager.

25   3.6   Agreement and Installation of New Rule

(Agree

    :sender SeniorManager

    :receiver Intermediary

    :content :ActionContent :Effect VerifyRule only give Fred days off

30

        :PreconditionCont nt

          if (Fred = employee && request = dayoff)

          dayoff = tru ;

22

```
        else
            dayoff = false
    :id 3
    :context 2
5   )
```

The new rule is added to the rule set and an agreement message is sent back to the originating manager.

It is available to a manager to include a rule in the local rule set 7b which
10 triggers the intermediary 3 to send a message to the manager's interface 7 under certain conditions. For instance, if a resource is requesting more than a certain amount of downtime in a fixed period, or an operative is requesting more than a certain number of days off. The manager might then have the means to override the decision of the intermediary 3. A valuable aspect of this is that the manager
15 alternatively might be able to detect trends reflecting the status of available resources for carrying out tasks and take appropriate action, such as bringing additional or substitute resources online.

Using the framework of messages between the interfaces 5,7,8, work co-ordination applications can be built. These not only allow control at multiple levels of
20 authority, but also inform people of what is happening, why it is happening, and how they can help other members of their team. Thus use of the system to support operatives also provides support for team building and social interaction processes in a distributed work team.

In the example given above, the availability data relates to operators. The
25 system can equally well be used to support changes in machine or processing capacity. For instance, the OSS might load the system by providing data of the following type:

4.   Setting up
30 **4.1   Equipment details and capacity status**

```
    Eg        EquipmentInfo
              {Computer: 143.199.123.456          computer address
              Mod I d  scription: desktop pc       s rver | desktop pc
```

ProcSpeedmaxstatus: 250MHz	0MHz | 250MHz | 500MHz
Modemmaxstatus: 8Kbits/s	8Kbits/s | 32Kbits/s | 64Kbits/s
Protocolstatus: ISDN;MIDI	ISDN | MIDI
RAMmaxstatus: 32 Mbytes	32 Mbytes | 64 Mbytes
Hardmaxstatus: 7 Gbytes}	7 Gbytes | 14 Gbytes

5

This shows that for a desktop personal computer (pc) available to the system, the maximum processor speed available for applications to be run by th system is half the maximum available, the modem speed is badly reduced, the pc has

10   ISDN and MIDI capability, the random access memory (RAM) is already half full and so is the hard disc.

Again, it may also be necessary that time or date information is provided so that availability information can be mapped against time or date. This could be provided to the intermediary 3 by the OSS 1 but might better be provided by the

15   resource interfaces 5. (If the time or date information is provided by the resource interfaces 5, each resource interface need only send capacity data to th intermediary 3 with respect to the dates/times at which capacity on its resource is available to the system.) The intermediary 3 then selects data with respect to availability of its resources in the light of tasks to be performed and forms an initial

20   "confirmed availability model":

4.2   Confirmed model:

{ListOfEquipment{all EquipmentInfo}

MIDIat64Kbits/s = 3	sum of MIDI PCs available at full modem speed
ISDNat32Kbits/s = 15	sum of ISDN PCs available at half modem speed
PCsat500MHz = 4}	sum of PCs available at top processor speed

25

This particular availability model might be generated in the light of tasks involving downloading audio and music, and running major software applications.

30   In order for the system to be enabled to respond in use on behalf of specific machines, the resource interfaces 5 again need to be loaded with respective resource profiles 5b. This can be done manually but it could also b done from the "Equipment details and capacity status" provid d by the OSS 1. Hence the resourc

24

interface 5 for a specific PC may have a profile loaded for that PC which shows that the PC is MIDI enabled, or has enabling software loaded for running a particular application.

5   **5.**   **The method for reallocating capacity away from the system**

An operative may enter data to the resource interface 5 for a particular PC which indicates that the actual available modem speed of that PC is to be reduced, for instance because the same PC is to be used for downloading non-system data. Alternatively, the PC may itself generate the data as a result of technical changes

10   occurring, such as a fault. The resource interface 5 uses its processing software 5a and message handler 5e to translate this into the following agent message and sends it to the intermediary 3:

5.1   Request

15   (Request

      :sender 143.199.123.456Agent

      :receiver Intermediary

      :content :effect   143.199.123.456 is a MIDI PC available at full modem speed whose speed is to be reduced

20   :id 1

      :context 1

)

The intermediary takes the confirmed model (4.2 above), subtracts 1 from

25   the MIDIat64Kbits/s field and uses it as the proposed model:

5.2   Proposed model:

    {ListOfEquipment{all EquipmentInfo}

    MIDIat64Kbits/s = 2   sum of MIDI PCs available at full modem speed

30   ISDNat32Kbits/s = 15   sum of ISDN PCs available at half modem speed

    PCsat500MHz = 4}   sum of PCs available at top processor spe d

25

The intermediary 3 now checks the proposed model against rel vant local and global rules 7b, 3c. It may be that the system can tolerate losing a MIDI-enabled link, at least temporarily, as long as an ISDN connection can replace it, for instance because music destined for download over a MIDI link can be downloaded in a

5   different format over any ISDN link.

5.3.1 Rule:

if (ISDNat32Kbits/s + MIDIat64Kbits/s > 17)

MIDI = true

10   Else

MIDI = false

The proposed model fails the rule so the request is refused and the proposed model abandoned. A refuse message is sent to all the resource interfaces 5. One of

15   the resource interfaces may find, on looking at the resource profile it has stored against its resource, that the resource is MIDI-enabled and has a rule that, if a refuse message in respect of a MIDI enabled PC is received, it will withdraw from a non-system task so that it can effectively offer to add to the tally of "MIDIat64Kbits/s". Thus the resource interface for this resource will issue an offer message in much the

20   same way as "Melagent" offers overtime in the operative-based example above. The intermediary will be triggered to generate a new proposed model and this time it will be found acceptable. The intermediary 3 notifies all the resource interfaces 5 and th resource interface 5 for the MIDI enabled PC which originally requested a speed reduction will successfully resubmit the request.

25   It can thus be seen that the system can be used to co-ordinate availability of machine resources directly and in response to outputs of the resources in real time. Such a system, by using "refuse" and "offer" messages, effectively self-corrects within limits when capacity is lost.

A consequence of a refuse message in these circumstances might be that

30   the work tasks actually have to be completely abandoned in the case that some capacity is lost, because a set of interdependent tasks as a whole is not going to be achievable. A system according to an embodiment of the present invention provides the advantage that this situation is effectively registered and a system can for

26

instance close itself down gracefully and immediately in that event rather than pointlessly completing only some of the necessary tasks.

Referring to Figures 2, 5 and 6, in embodiments of the present invention, if a non-programmer is to control the behaviour and functionality of a software system

5 effectively, the GUI preferably provides a highly responsive style of interaction (Burnett, Baker, and van Zee 1995), in a manner similar to a spreadsheet.

The object is not so much the "window style" (in the sense of how it appears to a user), but rather the idea of Visual Processing Languages (VPL). The goals of VPL designers are to improve the programmer's ability to express program logic and

10 to understand how the program works. These goals cannot be realised simply by switching from text to pictures. Instead the VPLs employed in embodiments of the present invention employ visual techniques to achieve one or more of the following preferable characteristics:

1. Fewer concepts required to program. For example, in many VPLs the

15 programmer does not deal with pointers, storage allocation, declarations, scope or variables.

2. Concrete programming process. In many VPLs the programmer can se , explore and change specific data values or even sample executions.

3. Explicit depiction of relationships. Constraint and dataflow diagrams are

20 example techniques.

4. Immediate visual feedback. After a program edit, in many VPLs immediate display of updated results helps the programmer find errors sooner.

The system of the invention preferably provides immediate feedback regarding any effects of user actions, without going through a "compile" step. Nardi (Nardi

25 1993) argues that, apart from immediate feedback, the system should communicate with the user in a language that is both task-oriented and visual. The power of such a language is its ability to externalise users' mental models of the control task (Mehandjiev 1997)

Control at different levels of responsibility use different visual representations.

30 For example there will be substantial differences between the GUI 5c for a resource interface 5, which can be constructed as a form with a limited set of work preferences, and the visual language employed at the GUI 7c, 8c for a manager's interface 7a, 8a to control local rules and conflict r solution policies. This is similar

to the layered approach to visual software control and visual programming advocated by Repenning and Ioannidou (1997). Whereas they use programming competency as the factor to determine the type of representations at each layer. However, the present invention preferably uses representations in a task-specific language which

5   are determined by the type of task performed by people at different organisational levels.

At higher levels of managerial control the complexity and variety of control tasks will increase to an extent where a single representation is insufficient. A person will have to interact with several different representations depending on the

10   type of control task they perform. A manager, for example, will need one type of representation to control business rules and another type of representation to control the team composition and preferences regarding type of work.

Facilities for integrating different representations into a unified control language with a high level of responsiveness are an important feature of this approach to

15   building work co-ordination software. A worker based embodiment of the present invention is now described from the point of view of the representations available to the_users, particularly the managers.

It deals with the domain of job scheduling for Customer Service Teams. Workers can request days off, overtime, and specify preferred task difficulty. They

20   do this through a single representation, but the system is designed so that other representations can be added. Also, different representations can be used for different types of hardware, for example a worker can use a mobile phone, a palm-top computer or PC to interface with the system.

The team manager can control scheduling rules and constraints, and monitor

25   the current team composition in terms of preferred task type, task difficulty, overtime and days off. Referring to Figure 6, in the example, the manager uses three different representations, which are tightly integrated together so that changes on one representation can immediately be seen on the screen in the other representations, and also notifications from the Intermediary.

30   The system keeps users informed of the outcome of their and others' requests following the scenarios described arli r. It specifies reasons for rejection and suggests actions to help others.

28

The system supports two main modes of communication: a mode of synchronous visual interaction between users and their resource interfaces, and asynchronous agent-based message passing and negotiation between the resource interfaces and the Intermediary. This is reflected by the components of the resource

5   interfaces 5 and the intermediary 3 shown in Figures 3 and 4.

Users of the system (workers or managers) will interact using GUIs 5c specialized for the purposes of different control tasks. The user interactions with each window will be translated to commands for instance for changing work coordination preferences, which can be understood by the resource interface 5. The

10   translation for each GUI 5c will be performed by a software module called "Translator". Different pairs (GUI, translator) can be installed or selected for running to provide different representations and different levels of control at the GUIs 5c.

The resource interface 5 is a software agent which provides support for an individual user of the system, and maintains a local "Model" of their work and team

15   preferences, including a list of "friends". The manager and trusted third party interfaces 7,8 will have a different functionality than the resource interfaces 5 to provide a higher level of control.

Each interface 5,7,8 communicates with the "Intermediary" 3 using a FIPA-compliant wrapper (FIPA98) and ACL (agent communication language) as a standard

20   language.

In different embodiments, the complexity of the proposed model and the process of determining if it is viable will vary. In some, the proposed model may be created by simple data transformations.

A simple example of forming a model, and of determining whether it is viable

25   (i.e. is compatible with constraint data defining the workload), is for the model to have a register indicating "the number of workers taking overtime". If a worker requests overtime then in the proposed model "the number of workers taking overtime" is one greater than "the number of workers taking overtime" in the confirmed model. Similarly, the model may have a register recording the number of

30   workers asking for holiday.

The test of whether this is a model consistent with constraints may then just be "is the number of people taking overtime at least twice the number of people asking

29

for holiday?". In other words, the intermediary 3 employs a relatively simple threshold.

In more complex embodiments (for example in situations in which there are more complex constraints set by a local manager or the OSS), the mechanism for

5 generating models and/or testing the models against constraints may be more complex, for example involving an expert system such as the classic DENDRAL expert system, as described in the paper "DENDRAL and META: the application dimensions" by Buchanan and Feigenbaum. DENDRAL was directed to "mechanizing" scientific reasoning using artificial intelligence techniques, and in

10 particular the generation of scientific hypotheses consistent with predetermined knowledge and experimental observations. The method included proposing candidate models (that is hypotheses) and testing whether they were consistent with th knowledge and experimental observations (which thus acted as constraints on th search space). DENDRAL included a module CONGEN which performed this task.

15 These algorithms can be adapted for use in the present invention, for example with management rules playing the role of scientific knowledge and inputs by operators playing the role of experimental observations. The proposed model in the present invention thus broadly corresponds to the proposed hypothesis of the DENDRAL approach.

20 The approach presented here can be generalized in several directions. The architecture is capable of supporting different interfaces, so that for example touch-tone telephones can be used by customers to negotiate service times.

Differentiation between the different levels and types of control can be done at the level of the local-resource interface 5, so managers at different levels of authority

25 may be connected to the system. For example higher levels of management may control scheduling policies and rules.

A domain in which embodiments of the present invention might usefully be applied is that of call centres. For example, the system can enable call centre staff to control call routing preferences and participate in scheduling their on-job training.

30

Ref renc s

The following documents are referred to in the text above. Their disclosure is inc rporated her in by r ference.

30

Buchanan and Feigenbaum (1978), DENDRAL and META-DENDRAL: their applications dimension, Artificial Intelligence, vol. 11, p 5-24.

Burnett, M. M., M. J. Baker, and P. van Zee (1995, March). Scaling up visual programming languages. IEEE Computer 28(3), 45.

5   Dourish, P., J. Holmes, A. MacLean, P. Marqvardsen, and A. Zbyslaw (1996, November). Freeflow: Mediating between representation and action in workflow systems. In ACM Computer Supported Cooperative Work. ACM.

FIPA98 1998, FIPA 98 Specifications, http:\\www.fipa.org\

Jennings NR, Faratin P, Johnson MJ, O'Brien P, and Wiegand ME. 1996. Using

10   intelligent agents to manage business processes. In Proceedings of the First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96), London, UK. The Practical Application Company Ltd, pp 345-360.

Mehandjiev, N. (1997). User Enhanceability for Information Systems through

15   Visual Languages. PhD thesis, Department of Computer Science, University of Hull, HULL HU6 7RX, England.

Nardi, B. A. (1993). A Small Matter of Programming: Perspectives on End Us r Computing. Cambridge, Massachusetts, USA: Massachusetts Institute of Technology.

20   Repenning, A. and A. Ioannidou (1997, September). Behavior processors: Layers between end-users and Java virtual machines. In Proceedings of VL'97. IEEE.

Rocco, E. (1998). Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact, Proceedings of CHI 98, April 1998.

Sachs, P. (1995, September). Transforming work: Collaboration, learning, and

25   design. Communications of the ACM 38(9), 3644.

Swenson, K et al. A business process environment supporting collaborativ planning. Journal of Collaborative Computing, 1(1): 15;34, March 1994.

31

CLAIMS

1. A method of supporting resource allocation for use in carrying out respective tasks which together fulfill one or more work requirements, each resource being provided with a resource interface, the method comprising:

    storing constraint definition data defining constraints on said allocation of resources;

    storing an initial data representation of resource availability;

    receiving at data processing means, from at least one of said resource interfaces, availability data concerning availability of a resource;

    generating a proposed-data representation of resource availability, based on the initial data representation together with said availability data;

    determining whether said proposed data representation is compatible with said constraint definition data;

    in the case that said proposed data representation is compatible with said constraint definition data, substituting said proposed data representation for said initial data representation to generate a new initial data representation; and

    in the case that said proposed data representation is not compatible with said constraint definition data, transmitting a rejection signal to said at least one of said resource interfaces.

2. A method according to Claim 1 which further comprises:

    receiving at the data-processing means, from at least one of said resource interfaces, further availability data concerning availability of a resource;

    generating a further proposed data representation of resource availability, based on the initial data representation together with said further availability data;

    determining whether said further proposed data representation is compatible with the constraint definition data;

    in the case that said further proposed data representation is compatible with said constraint definition data, substituting said further proposed data

32

representation for said initial data representation to generate a new initial data representation; and

in the case that said proposed data representation is not compatible with said constraint definition data, generating and transmitting a rejection signal to said

5   at least one of said resource interfaces.

3.   A method according to either one of the preceding claims, wherein, in the case that a proposed data representation is not compatible with said constraint definition data, the step of generating and transmitting a rejection

10   signal to said at least one of said resource interfaces comprises generating and transmitting a rejection signal to a plurality of said resource interfaces.

4.   A method according to any one of the preceding claims wherein at least one resource interface is provided with at least one resource profile, the resource

15   profile comprising data in respect of a resource, the method further comprising the steps of:

receiving at a resource interface a rejection signal;

reviewing a resource profile provided with respect to that resource interface; and

20   outputting availability data to the data processing means dependent on the outcome of the review.

5.   A method according to Claim 4 wherein said resource profile comprises at least first and second data types in respect of a resource, the first data type

25   comprising at least one resource attribute and the second data type comprising availability commitments of the resource.

6.   A method according to either one of Claims 4 or 5 wherein the resource profile further comprises a priority indicator for at least one availability commitment

30   of the resource, and wherein said step of reviewing a resource profile compris s r viewing the priority indicator.

33

7. A method according to any one of Claims 4, 5 or 6 wherein said rejection signal comprises an identifier for a selected resource, or for a s lected set of resources, and wherein said steps of reviewing a resource profile and outputting availability data to the data processing means dependent on the

5   outcome of the review comprise reviewing the resource profile for the presence of said identifier and outputting availability data only if said identifier is present.

8. A method according to any one of the preceding claims which further

10   comprises, subsequent to generating and transmitting said rejection signal, triggering-termination-of-tasks-being carried out-in-respect-of-a common-work requirement to which the rejection signal is related.

9. A method according to Claim 8 wherein said step of triggering termination is

15   carried out after a predetermined time has elapsed during which no availability data has been received from a resource interface.

10. A method according to any one of the preceding claims wherein   said constraint definition data comprises at least two sets of constraint definition

20   data, and the method further comprises:

receiving via a user interface a proposed modification to a first set of constraint definition data;

reviewing the proposed modification against the second set of constraint definition-data;

25   in the case that the proposed modification is compatible with the second set, modifying the first set accordingly; and

in the case that the proposed modification is not compatible with the second set, transmitting a rejection signal to the user interface.

30 11. Apparatus for supporting resource allocation in carrying out respective tasks which together fulfill one or more work requirements, the apparatus comprising:

at least one resource interface;

34

a constraint definition data store for storing data defining constraints on said allocation of resources;

a data representation store for storing an initial data representation of resource availability and a proposed data representation of resource availability; and

5   data processing means

the apparatus being adapted, in use, to maintain an initial data representation in the data representation store, to receive an input from at least one resourc interface comprising availability data concerning availability of a resource, to generate a proposed data representation of resource availability, to review the

10   proposed data representation of resource availability against the constraints, and either to substitute the proposed data representation of resource availability for the initial data representation or to output a rejection message to at least one resource interface, dependent on the outcome of said review.

15   12. Apparatus according to Claim 11 wherein said constraint definition data store comprises means for storing at least two sets of constraint definition data, each set having at least one respective input, said apparatus having means for reviewing constraint data received at one input against constraint data received at another input, and means for either outputting a rejection message

20   or for loading the received constraint data, in dependence on the outcome of the review.

13. Apparatus according to either one of claims 11 or 12 wherein each resource interface is provided with a profile store for storing at least one resource

25   profile and, on receipt of a rejection message, each resource interface is adapted to review any resource profiles stored in its profile store and, in the event that a resource profile is identified as relevant to the rejection message, to transmit to the data processing means a signal containing availability data for the respective resource.

30

14. Apparatus according to Claim 13 wherein a resource profile comprises at least one data element and a rejection message comprises at least one data element, review of a resource profile comprising matching the data element

35

from a rejection message against the data element or elements in a resource profile.
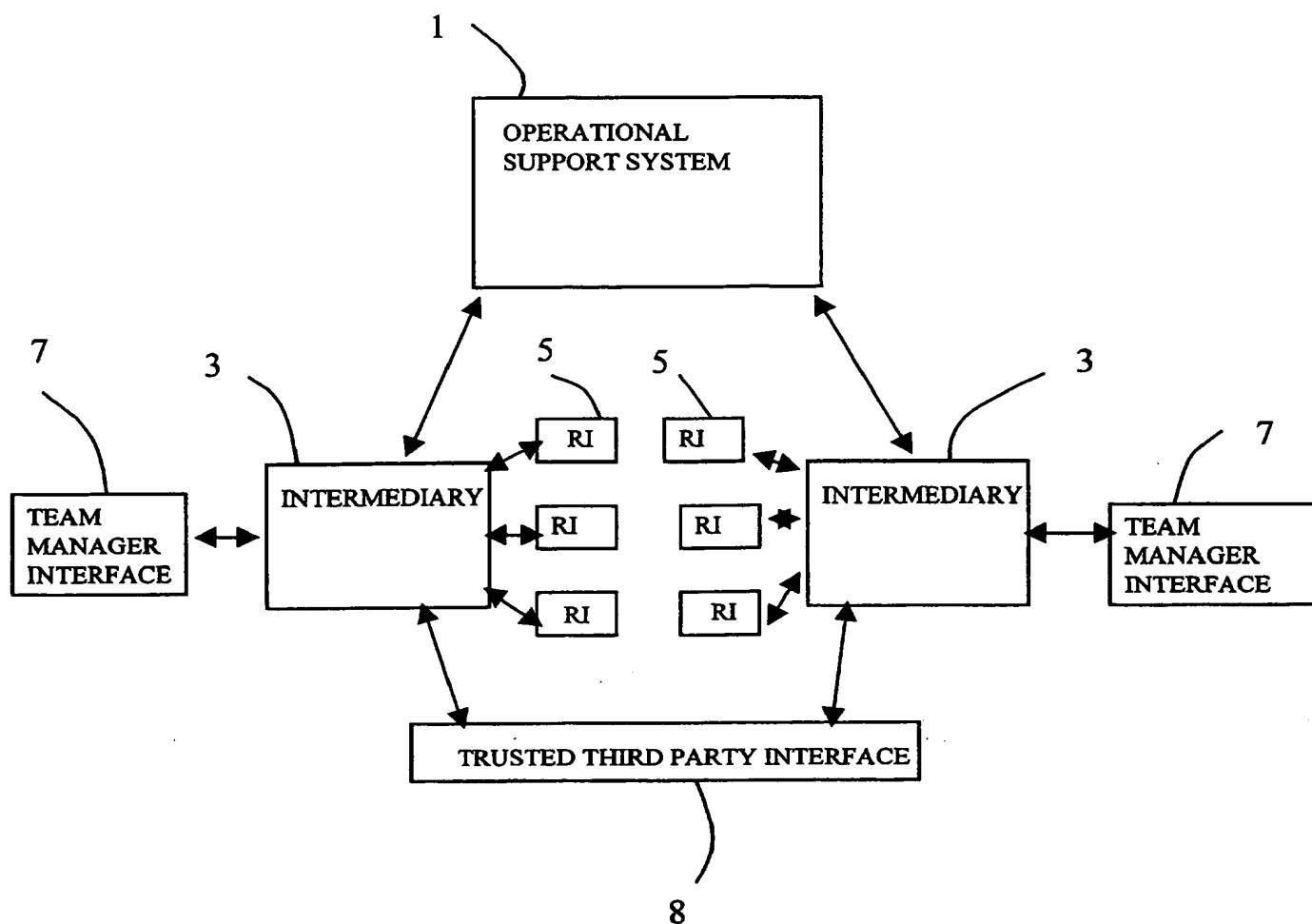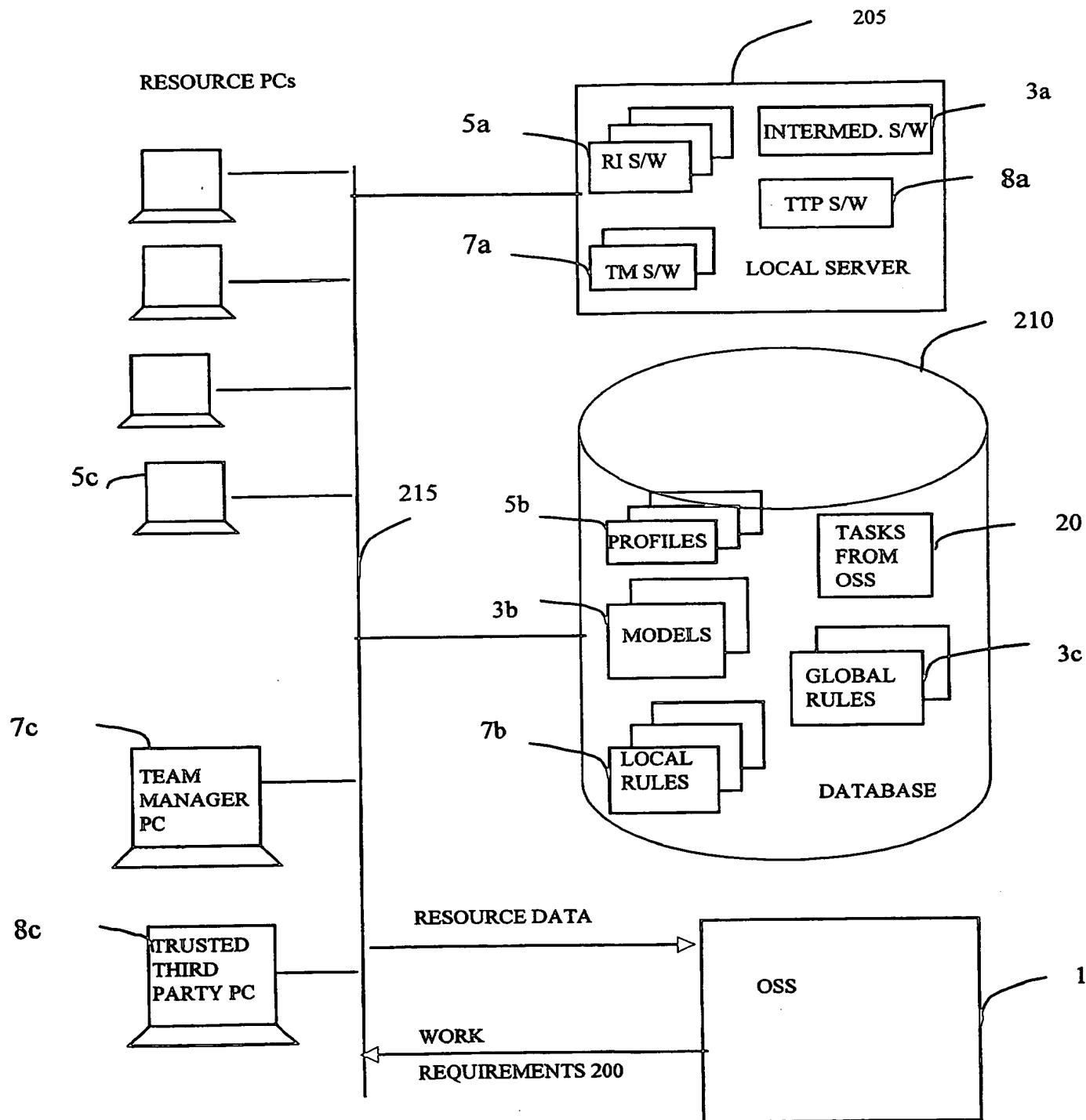
THIS PAGE BLANK (USPTO)

36

Abstract

Methods and systems are proposed for coordinating tasks carried out by operatives. An operational support system 1 defines requirements for work to be
5   carried out by operatives. The operatives receive instructions via an intermediary coordinator 3. The operatives are provided with a software agent 5, which empowers them to make requests. The intermediary 3 reconciles the work requirements with the requests. The method and systems use agent-based negotiation strategies and allow workers and team managers to control the system in a visual interactive
10   fashion. The system can be used to enable workers to set work preferences, trade jobs, share knowledge as well as build informal alliances to help each other with their work. Managers are able to review and control local business rules and scheduling preferences using their own software agent 7.

15

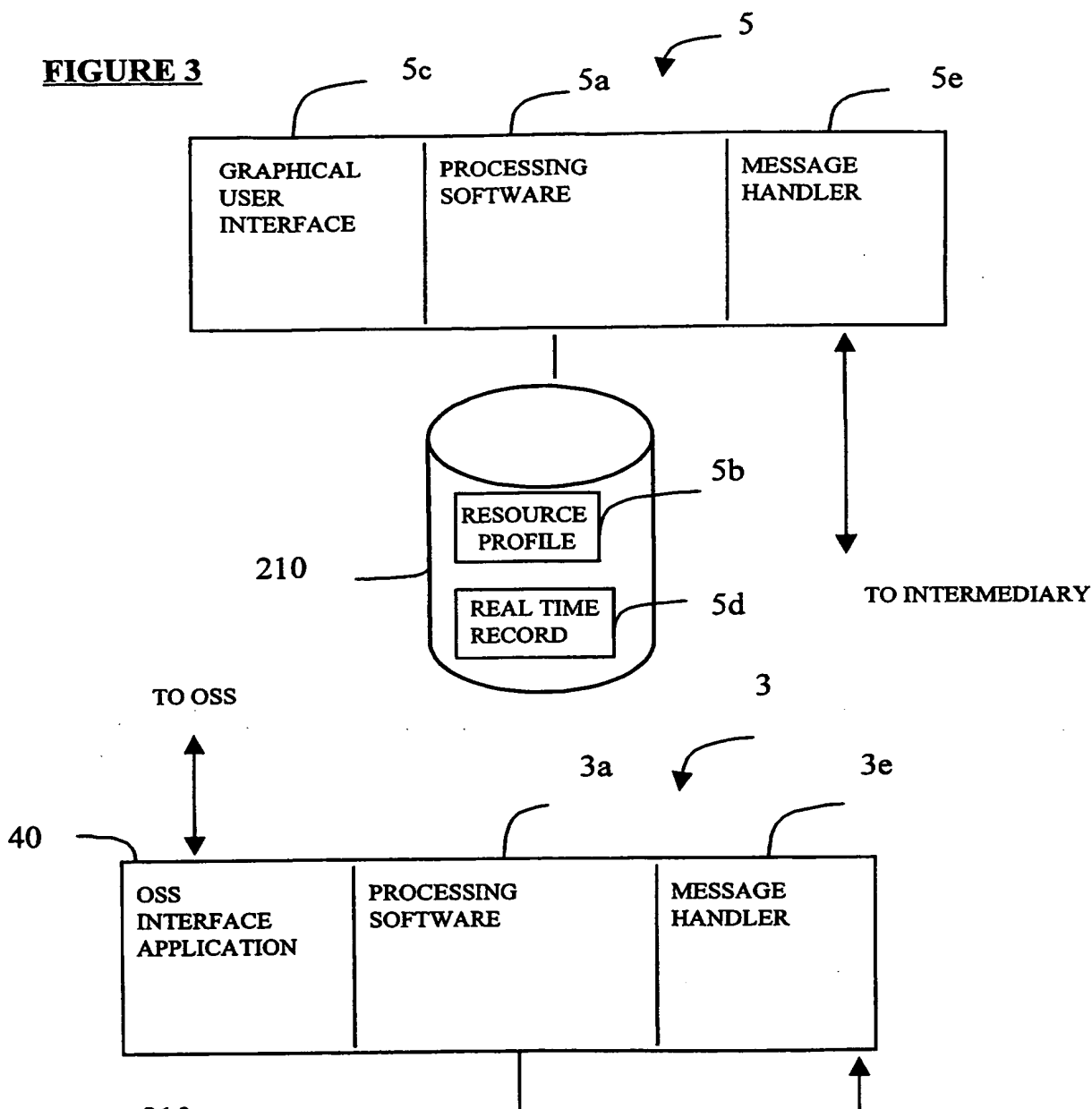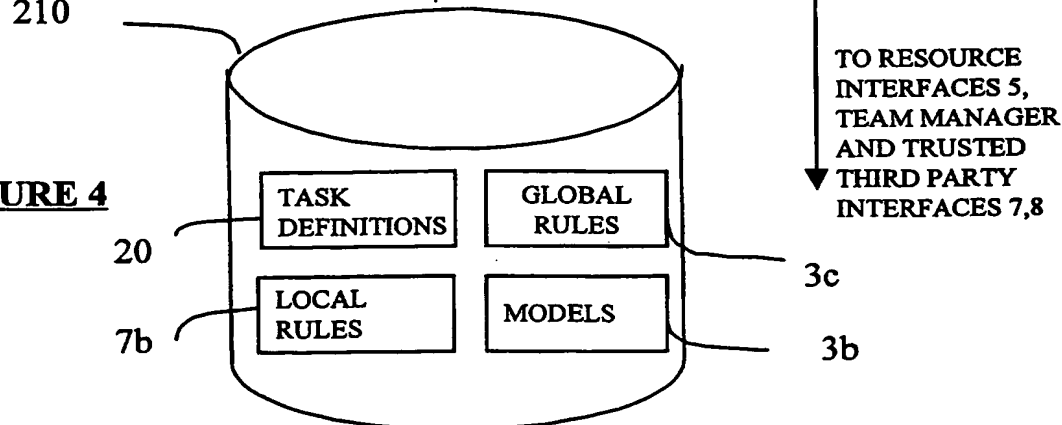THIS PAGE BLANK (USPTO)

1

OPERATIONAL
SUPPORT SYSTEM

7   3   5   5   3   7

| RI | | RI |

TEAM
MANAGER
INTERFACE

INTERMEDIARY

| RI | | RI |

INTERMEDIARY

TEAM
MANAGER
INTERFACE

| RI | | RI |

TRUSTED THIRD PARTY INTERFACE

8

FIGURE 1

RESOURCE PCs

205

5a

INTERMED. S/W — 3a

RI S/W

TTP S/W — 8a

7a

TM S/W   LOCAL SERVER

210

5c

215

5b

PROFILES

TASKS FROM OSS — 20

3b

MODELS

GLOBAL RULES — 3c

7c

TEAM MANAGER PC

7b

LOCAL RULES   DATABASE

8c

TRUSTED THIRD PARTY PC

RESOURCE DATA

OSS — 1

WORK REQUIREMENTS 200

FIGURE 2

FIGURE 3

5

GRAPHICAL USER INTERFACE	PROCESSING SOFTWARE	MESSAGE HANDLER

5c   5a   5e

210

RESOURCE PROFILE — 5b

REAL TIME RECORD — 5d

TO INTERMEDIARY

TO OSS

3

3a   3e

40

OSS INTERFACE APPLICATION	PROCESSING SOFTWARE	MESSAGE HANDLER

210

FIGURE 4

TASK DEFINITIONS	GLOBAL RULES
LOCAL RULES	MODELS

20

7b

3c

3b

TO RESOURCE INTERFACES 5, TEAM MANAGER AND TRUSTED THIRD PARTY INTERFACES 7,8
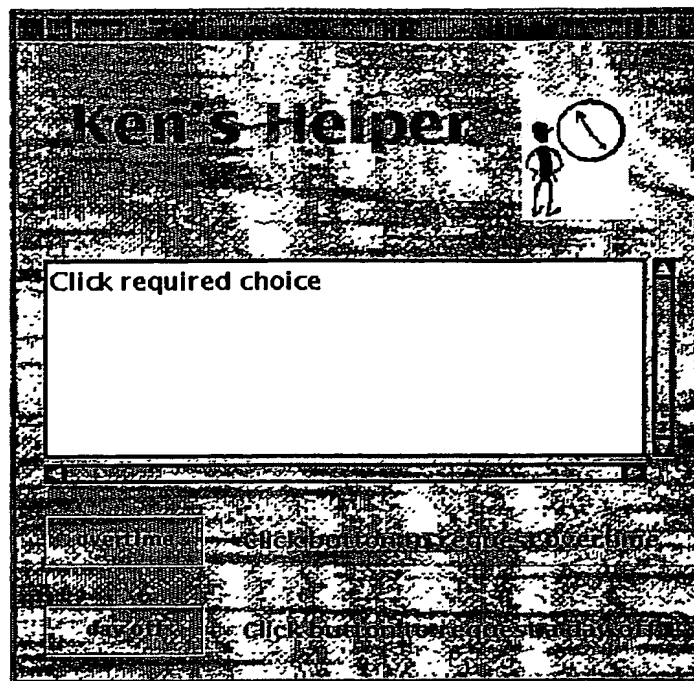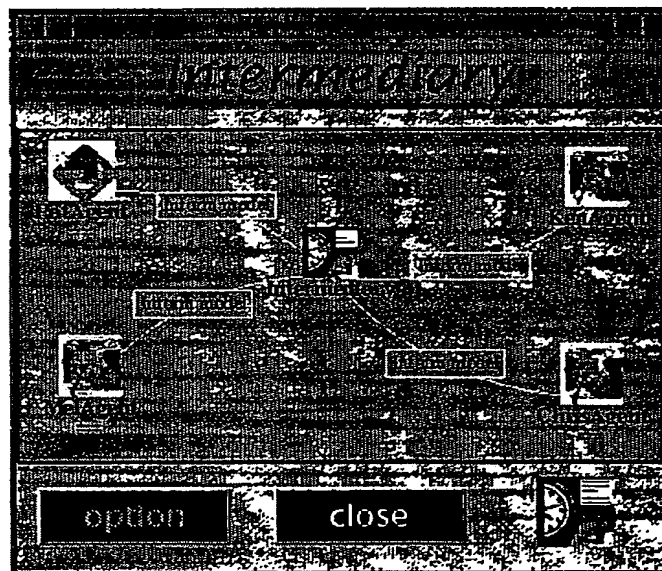
Figure 5: User Interface for PC user

Figure 6: Intermediary Information